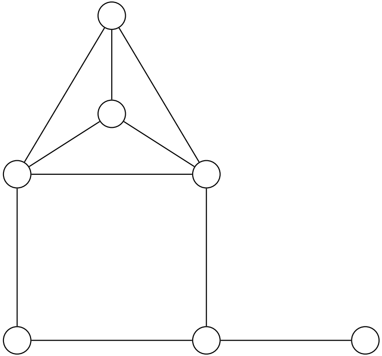
Algorithmics of Dynamic Well-Structured Graphs

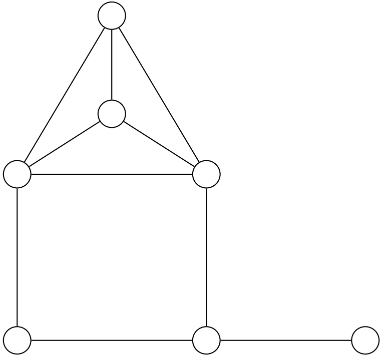
Marek Sokołowski

16 October 2025

Graphs & Graph problems

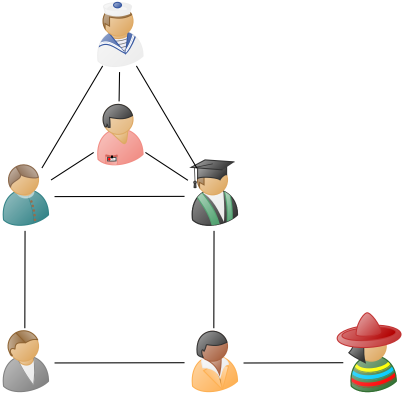


Graphs & Graph problems



n vertices, m edges

Graphs & Graph problems



n vertices, m edges

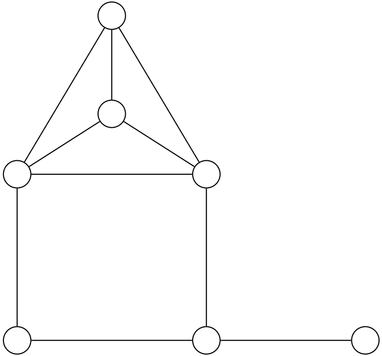
people **relationships**

Graphs & Graph problems

n vertices, m edges

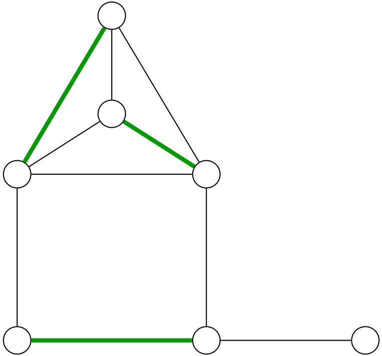
intersections **streets**

Graphs & Graph problems



n vertices, m edges

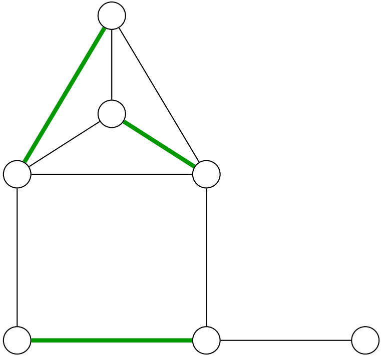
Graphs & Graph problems



n vertices, m edges

MAXIMUM MATCHING

Graphs & Graph problems



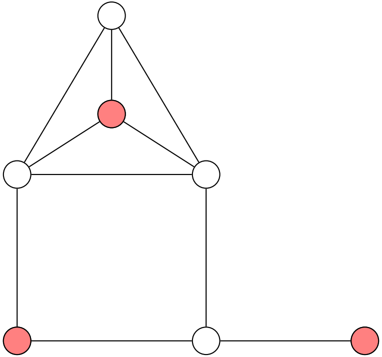
n vertices, m edges

MAXIMUM MATCHING

Easy!

[Edmonds '61]

Graphs & Graph problems



n vertices, m edges

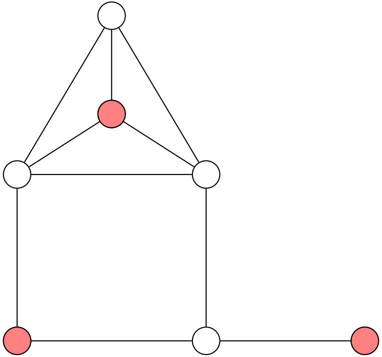
MAXIMUM MATCHING

Easy!

[Edmonds '61]

MAXIMUM INDEPENDENT SET

Graphs & Graph problems



n vertices, m edges

Maximum Matching

Easy!

[Edmonds '61]

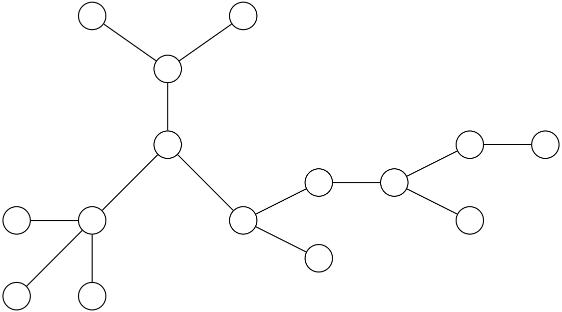
Maximum Independent Set

NP-hard!

[Cook '71, Karp '72, Levin '73]

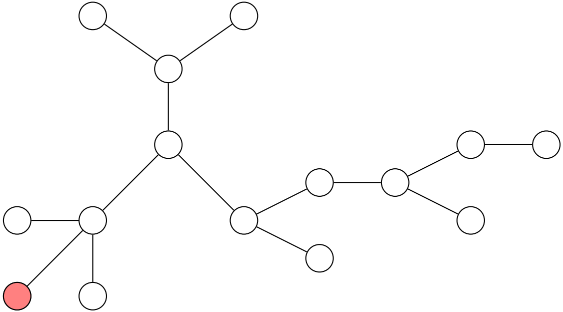
Trees

Maximum Independent Set is NP-hard in general... But becomes easy on **trees**!



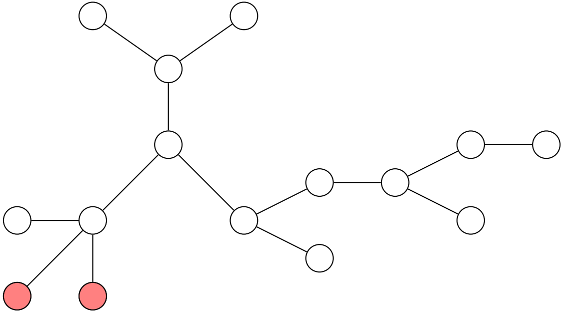
Trees

MAXIMUM INDEPENDENT SET is NP-hard in general... But becomes easy on **trees**!



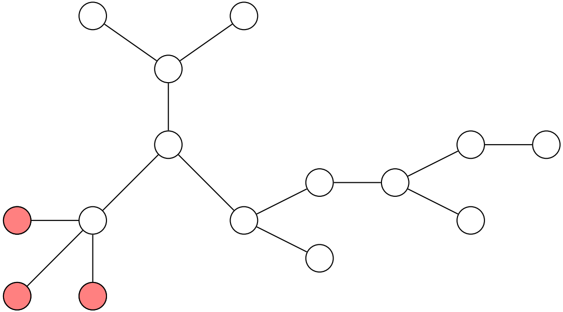
Trees

Maximum Independent Set is NP-hard in general. . . But becomes easy on **trees**!



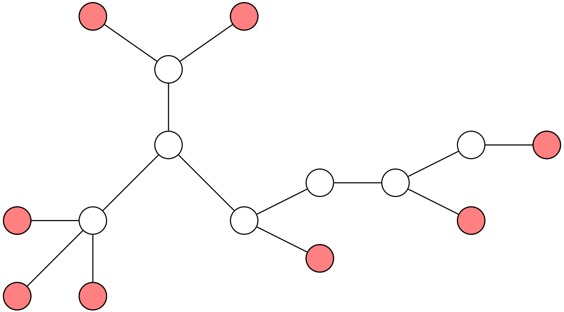
Trees

MAXIMUM INDEPENDENT SET is NP-hard in general... But becomes easy on **trees**!



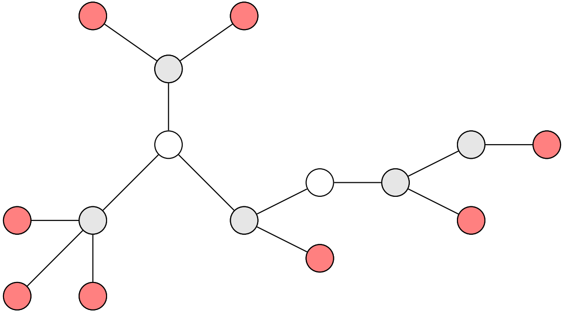
Trees

Maximum Independent Set is NP-hard in general... But becomes easy on **trees**!



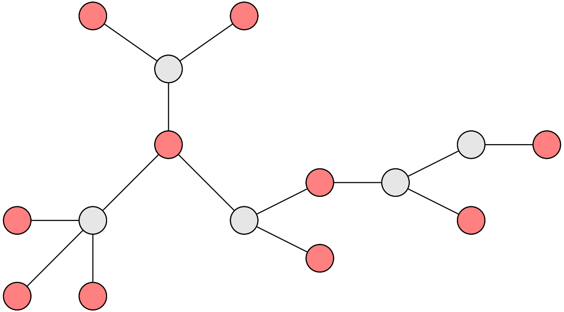
Trees

Maximum Independent Set is NP-hard in general... But becomes easy on **trees**!



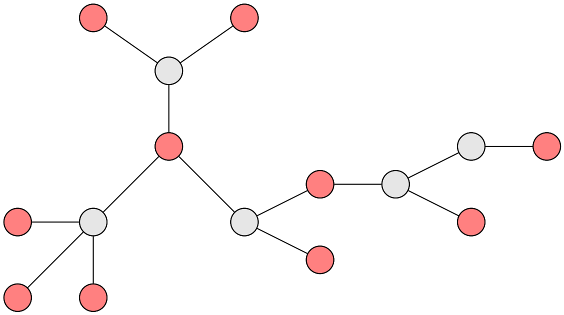
Trees

Maximum Independent Set is NP-hard in general... But becomes easy on **trees**!



Trees

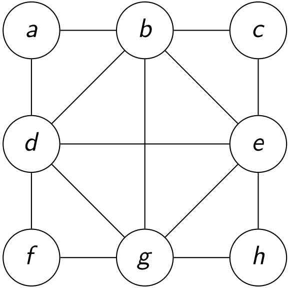
Maximum Independent Set is NP-hard in general... But becomes easy on **trees**!



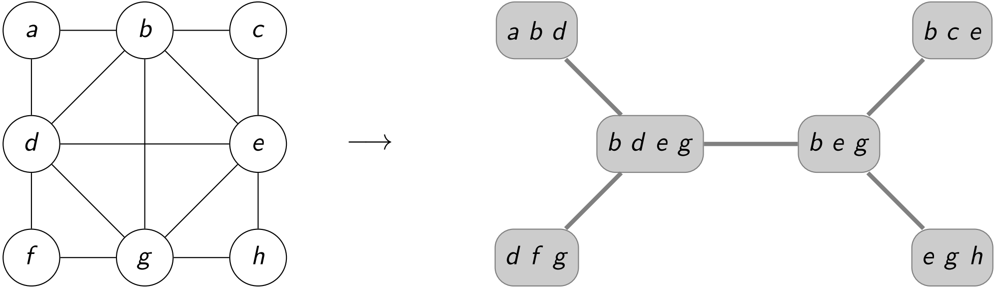
Question

Maybe some hard problems can be solved efficiently on **more general tree-like** graphs?

Treewidth

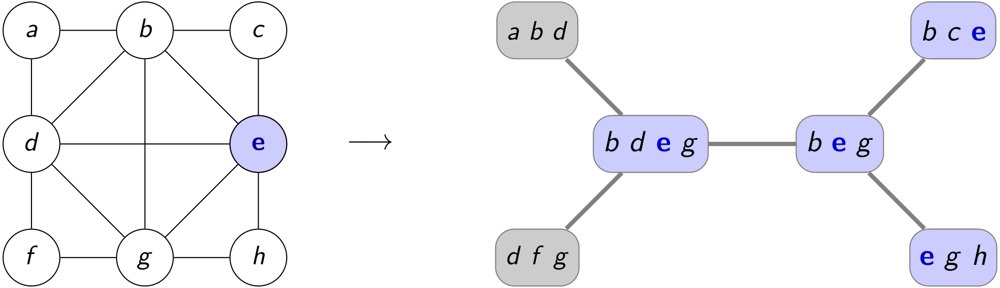


Treewidth



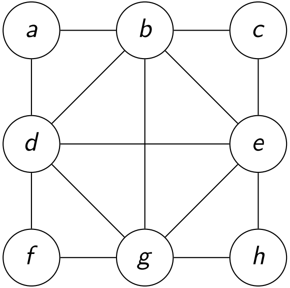
tree decomposition

Treewidth

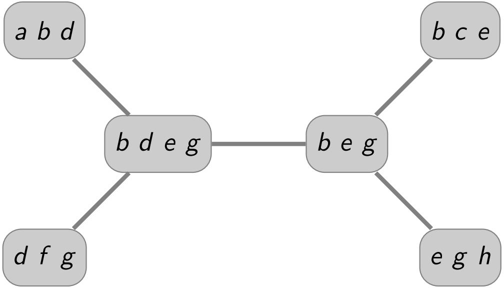


- Each **vertex** in a non-empty connected subgraph of the decomposition

Treewidth

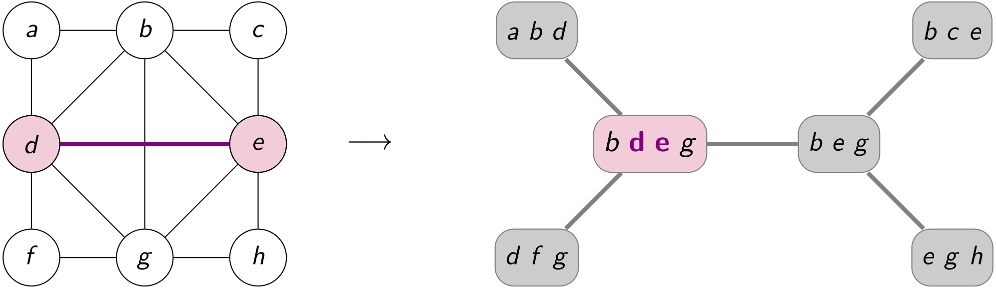


$\longrightarrow$



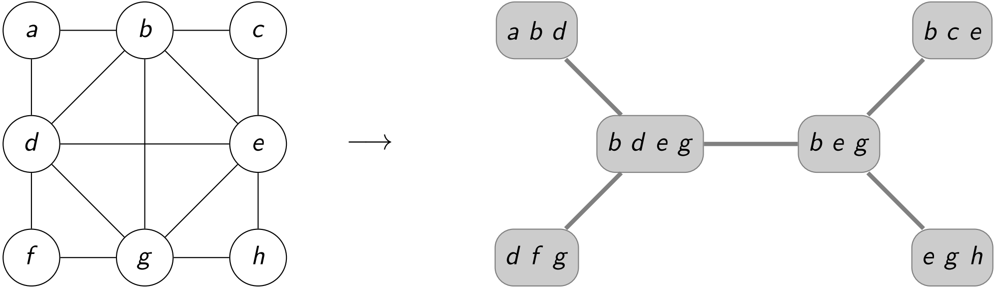
- Each **vertex** in a non-empty connected subgraph of the decomposition

Treewidth



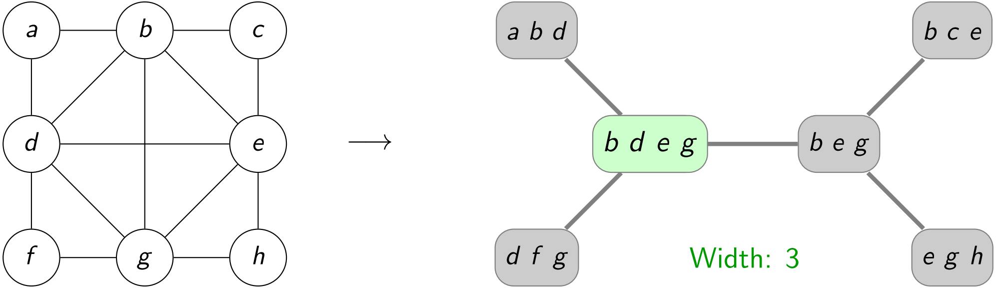
- Each **vertex** in a non-empty connected subgraph of the decomposition
- Each **edge** $uv \implies$ both u and v in some common bag of the decomposition

Treewidth



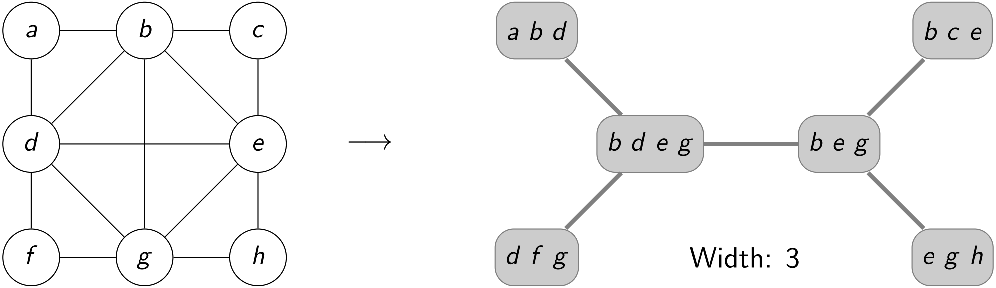
- Each **vertex** in a non-empty connected subgraph of the decomposition
- Each **edge** $uv \implies$ both u and v in some common bag of the decomposition

Treewidth



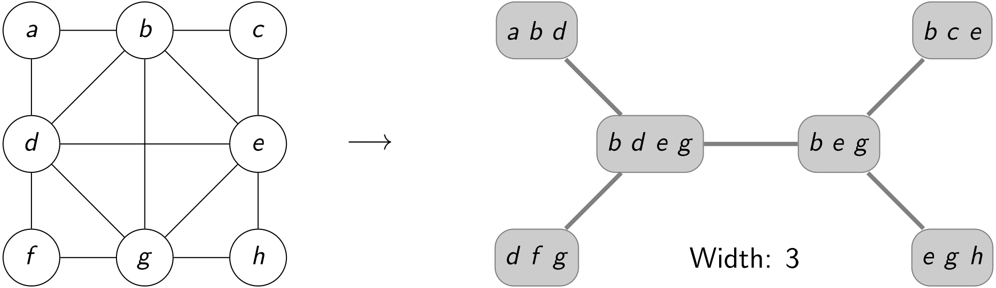
- Each **vertex** in a non-empty connected subgraph of the decomposition
- Each **edge** $uv \implies$ both u and v in some common bag of the decomposition
- **Width:** maximum bag size, minus 1

Treewidth



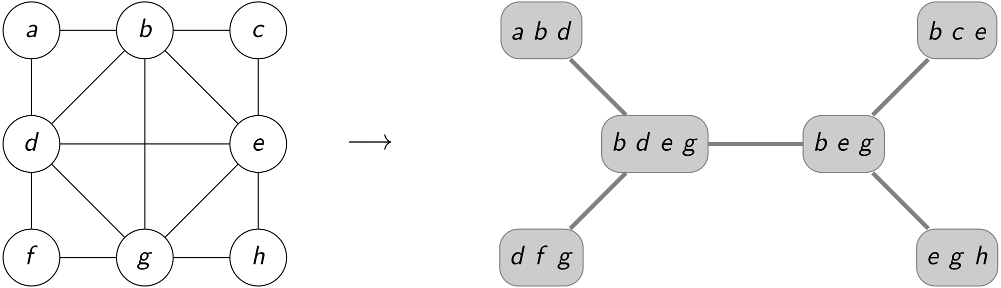
- Each **vertex** in a non-empty connected subgraph of the decomposition
- Each **edge** $uv \implies$ both u and v in some common bag of the decomposition
- **Width:** maximum bag size, minus 1

Treewidth



- Each **vertex** in a non-empty connected subgraph of the decomposition
- Each **edge** $uv \implies$ both u and v in some common bag of the decomposition
- **Width:** maximum bag size, minus 1
- **Treewidth:** minimum possible width of a tree decomposition

Treewidth



Treewidth is great!

Given: n-vertex graph G and its tree decomposition of width w

Then: Maximum Independent Set can be solved in time $2^{\mathcal{O}(w)} \cdot n$

Treewidth

Treewidth is great!

Given: n-vertex graph G and its tree decomposition of width w
Then: MAXIMUM INDEPENDENT SET can be solved in time $2^{\mathcal{O}(w)} \cdot n$

Problem: Usually we don't have a tree decomposition of a graph beforehand.

Treewidth

Treewidth is great!

Given: n-vertex graph G and its tree decomposition of width w
Then: Maximum Independent Set can be solved in time $2^{\mathcal{O}(w)} \cdot n$

Problem: Usually we don't have a tree decomposition of a graph beforehand.

Tree decomposition algorithms

Given an n-vertex graph G of treewidth w, we can **find** a tree decomposition of G...

Treewidth

Treewidth is great!

Given: n-vertex graph G and its tree decomposition of width w
Then: Maximum Independent Set can be solved in time $2^{\mathcal{O}(w)} \cdot n$

Problem: Usually we don't have a tree decomposition of a graph beforehand.

Tree decomposition algorithms

Given an n-vertex graph G of treewidth w, we can **find** a tree decomposition of G...

	Width guarantee	**Time**
[Robertson, Seymour '86]	$4w + 3$	$2^{\mathcal{O}(w)} \cdot n^2$

Treewidth

Treewidth is great!

Given: n-vertex graph G and its tree decomposition of width w
Then: MAXIMUM INDEPENDENT SET can be solved in time $2^{\mathcal{O}(w)} \cdot n$

Problem: Usually we don't have a tree decomposition of a graph beforehand.

Tree decomposition algorithms

Given an n-vertex graph G of treewidth w, we can **find** a tree decomposition of G...

	Width guarantee	**Time**
[Robertson, Seymour '86]	$4w + 3$	$2^{\mathcal{O}(w)} \cdot n^2$
[Bodlaender '96]	w	$2^{\mathcal{O}(w^3)} \cdot n$

Treewidth

Treewidth is great!

Given: n-vertex graph G and its tree decomposition of width w
Then: MAXIMUM INDEPENDENT SET can be solved in time $2^{\mathcal{O}(w)} \cdot n$

Problem: Usually we don't have a tree decomposition of a graph beforehand.

Tree decomposition algorithms

Given an n-vertex graph G of treewidth w, we can **find** a tree decomposition of G...

	Width guarantee	**Time**
[Robertson, Seymour '86]	$4w + 3$	$2^{\mathcal{O}(w)} \cdot n^2$
[Bodlaender '96]	w	$2^{\mathcal{O}(w^3)} \cdot n$
[Bodlaender et al. '16]	$5w + 4$	$2^{\mathcal{O}(w)} \cdot n$

Treewidth

Treewidth is great!

Given: n-vertex graph G and its tree decomposition of width w
Then: MAXIMUM INDEPENDENT SET can be solved in time $2^{\mathcal{O}(w)} \cdot n$

Problem: Usually we don't have a tree decomposition of a graph beforehand.

Tree decomposition algorithms

Given an n-vertex graph G of treewidth w, we can **find** a tree decomposition of G...

	Width guarantee	**Time**
[Robertson, Seymour '86]	$4w + 3$	$2^{\mathcal{O}(w)} \cdot n^2$
[Bodlaender '96]	w	$2^{\mathcal{O}(w^3)} \cdot n$
[Bodlaender et al. '16]	$5w + 4$	$2^{\mathcal{O}(w)} \cdot n$
[Korhonen '21]	$2w + 1$	$2^{\mathcal{O}(w)} \cdot n$

Treewidth

Treewidth is great!

Given: n-vertex graph G and its tree decomposition of width w
Then: MAXIMUM INDEPENDENT SET can be solved in time $2^{\mathcal{O}(w)} \cdot n$

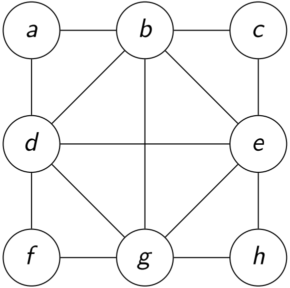
Problem: Usually we don't have a tree decomposition of a graph beforehand.

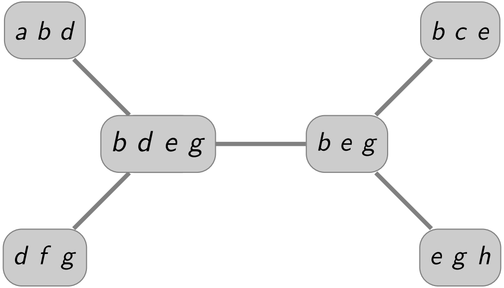
Tree decomposition algorithms

Given an n-vertex graph G of treewidth w, we can **find** a tree decomposition of G...

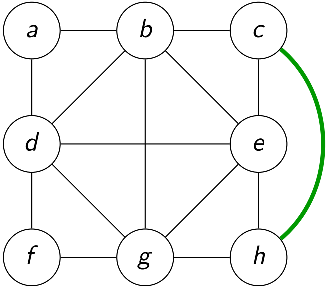
	Width guarantee	**Time**
[Robertson, Seymour '86]	$4w + 3$	$2^{\mathcal{O}(w)} \cdot n^2$
[Bodlaender '96]	w	$2^{\mathcal{O}(w^3)} \cdot n$
[Bodlaender et al. '16]	$5w + 4$	$2^{\mathcal{O}(w)} \cdot n$
[Korhonen '21]	$2w + 1$	$2^{\mathcal{O}(w)} \cdot n$
[Korhonen, Lokshtanov '23]	w	$2^{\mathcal{O}(w^2)} \cdot n$

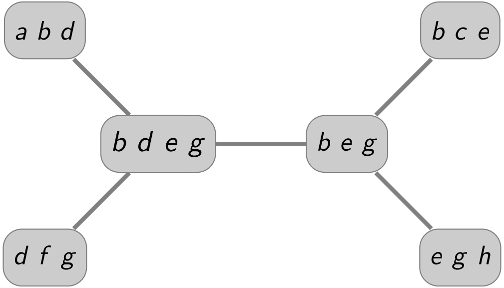
Suddenly. . .



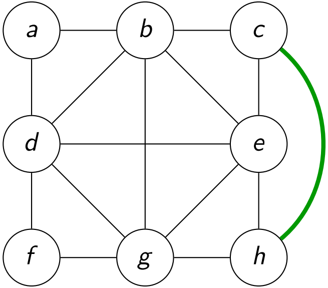


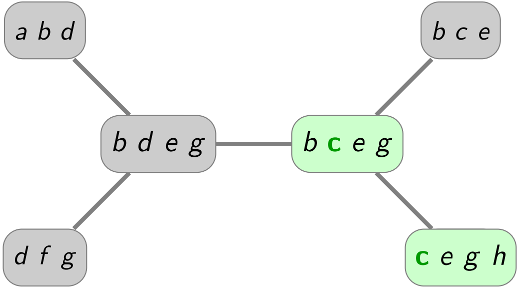
Suddenly. . .



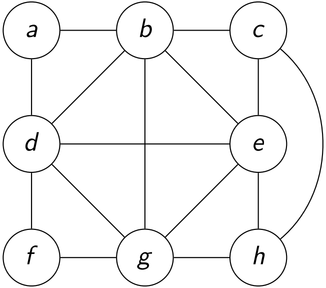


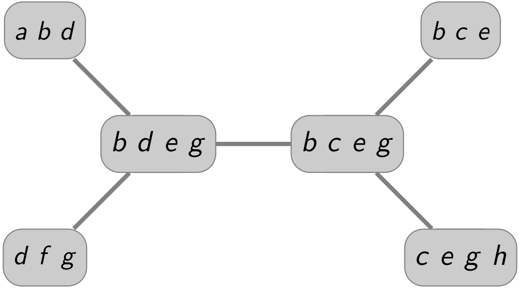
Suddenly. . .



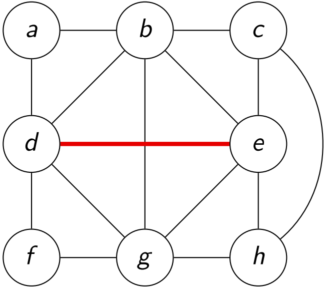


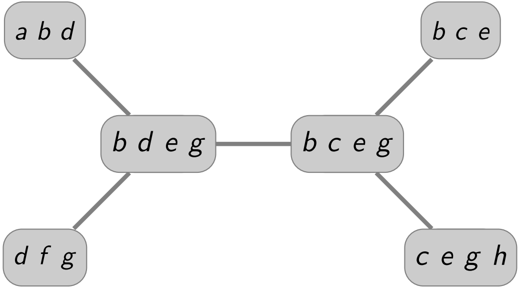
Suddenly. . .



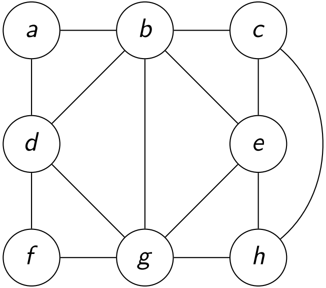


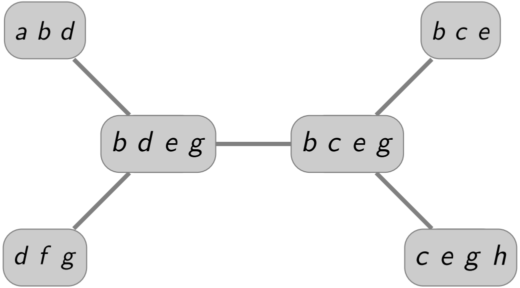
Suddenly. . .



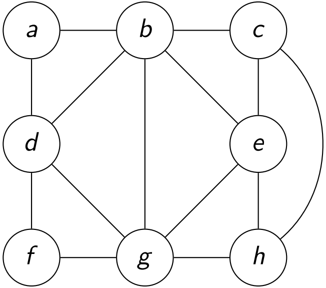


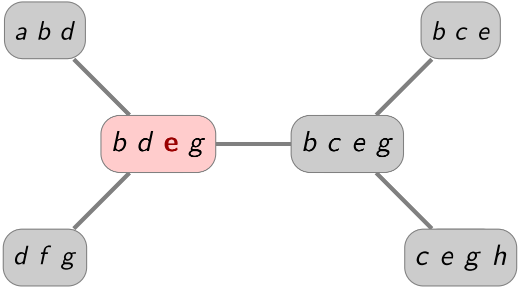
Suddenly. . .



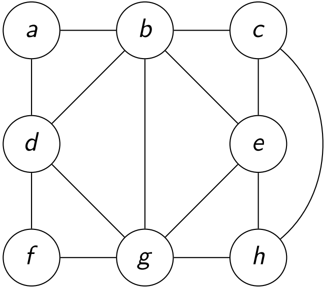


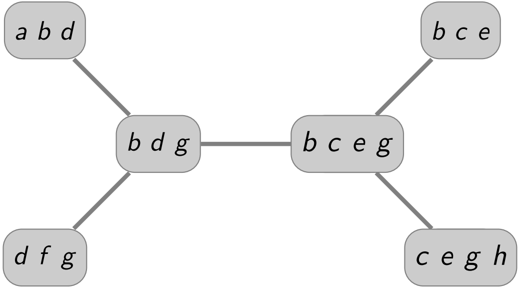
Suddenly...



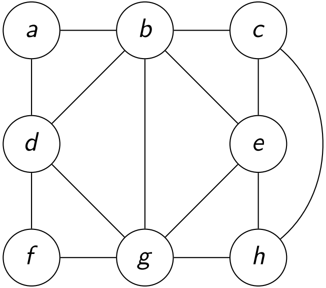


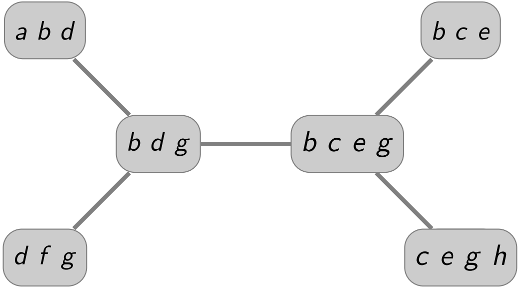
Suddenly. . .





Suddenly. . .





Problem

How to maintain tree decompositions of **dynamic graphs**?

Dynamic Treewidth

Korhonen, Majewski, Nadara, Pilipczuk, **Sokołowski** [FOCS '23]

Dynamic Treewidth

Main result

Dynamic Treewidth

Korhonen, Majewski, Nadara, Pilipczuk, **Sokołowski** [FOCS '23]

DYNAMIC TREEWIDTH

Main result

In a **dynamic graph** G with n vertices of treewidth $w \dots$

Dynamic Treewidth

Korhonen, Majewski, Nadara, Pilipczuk, **Sokołowski** [FOCS '23]

Dynamic Treewidth

Main result

In a **dynamic graph** G with n vertices of treewidth w ...

We maintain: a tree decomposition of G of width at most $6w + 5$ ...

Dynamic Treewidth

Korhonen, Majewski, Nadara, Pilipczuk, **Sokołowski** [FOCS '23]

Dynamic Treewidth

Main result

In a **dynamic graph** G with n vertices of treewidth $w \dots$

We maintain: a tree decomposition of G of width at most $6w + 5 \dots$

Initialization time: $2^{w^{\mathcal{O}(1)}} \cdot n$

Dynamic Treewidth

Korhonen, Majewski, Nadara, Pilipczuk, **Sokołowski** [FOCS '23]

DYNAMIC TREEWIDTH

Main result

In a **dynamic graph** G with n vertices of treewidth $w \ldots$

We maintain: a tree decomposition of G of width at most $6w + 5 \ldots$

Initialization time: $2^{w^{\mathcal{O}(1)}} \cdot n$

Update time: $2^{w^{\mathcal{O}(1)} \cdot \sqrt{\log n \log \log n}}$ *(amortized)*

Dynamic Treewidth

Korhonen, Majewski, Nadara, Pilipczuk, **Sokołowski** [FOCS '23]

Dynamic Treewidth

Main result

In a **dynamic graph** G with n vertices of treewidth $w \ldots$

We maintain: a tree decomposition of G of width at most $6w + 5 \ldots$

Initialization time: $2^{w^{\mathcal{O}(1)}} \cdot n$

Update time: $2^{w^{\mathcal{O}(1)} \cdot \sqrt{\log n \log \log n}}$ *(amortized)*

$$\log^{1000} n \quad \ll \quad \mathbf{2}^{\sqrt{\log \mathbf{n} \log \log \mathbf{n}}} \quad \ll \quad n^{0.001}$$

Dynamic Treewidth

Korhonen, Majewski, Nadara, Pilipczuk, **Sokołowski** [FOCS '23]

DYNAMIC TREEWIDTH

Main result

In a **dynamic graph** G with n vertices of treewidth w ...

We maintain: a tree decomposition of G of width at most $6w + 5$ ...

Initialization time: $2^{w^{\mathcal{O}(1)}} \cdot n$

Update time: $2^{w^{\mathcal{O}(1)} \cdot \sqrt{\log n \log \log n}}$ *(amortized)*

Extension

We can also dynamically solve any decision/optimization problem expressible in CMSO_2 logic.

Dynamic Treewidth

Korhonen, Majewski, Nadara, Pilipczuk, **Sokołowski** [FOCS '23]

DYNAMIC TREEWIDTH

Main result

In a **dynamic graph** G with n vertices of treewidth $w\dots$

We maintain: a tree decomposition of G of width at most $6w+5\dots$

Initialization time: $2^{w^{\mathcal{O}(1)}} \cdot n$

Update time: $2^{w^{\mathcal{O}(1)} \cdot \sqrt{\log n \log \log n}}$ *(amortized)*

Extension

We can also dynamically solve any decision/optimization problem expressible in CMSO_2 logic.

MAX MATCHING, MAX INDEPENDENT SET, LONGEST PATH, HAMILTONIAN CYCLE...

Dynamic Treewidth: Follow-Up

Korhonen [FOCS '25]

DYNAMIC TREEWIDTH IN LOGARITHMIC TIME

Follow-up result

In a **dynamic graph** G with n vertices of treewidth $w\ldots$

We maintain: a tree decomposition of G of width at most $9w + 8\ldots$

Initialization time: $2^{\mathcal{O}(w)} \cdot n$

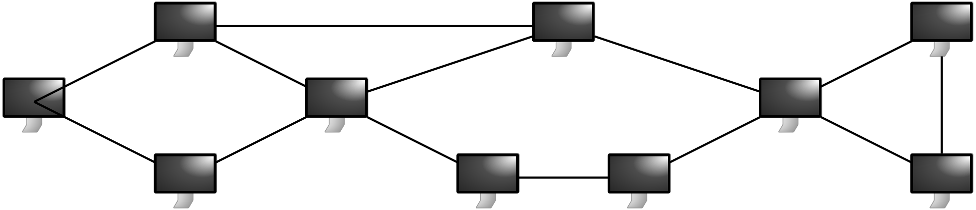
Update time: $2^{\mathcal{O}(w)} \cdot \log n$ *(amortized)*

Extension

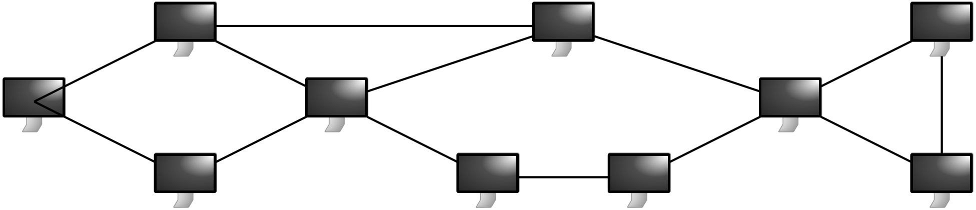
We can also dynamically solve any decision/optimization problem expressible in CMSO_2 logic.

MAX MATCHING, MAX INDEPENDENT SET, LONGEST PATH, HAMILTONIAN CYCLE...

Biconnectivity

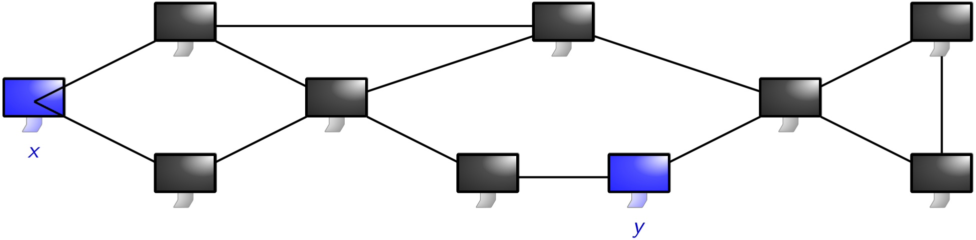


Biconnectivity



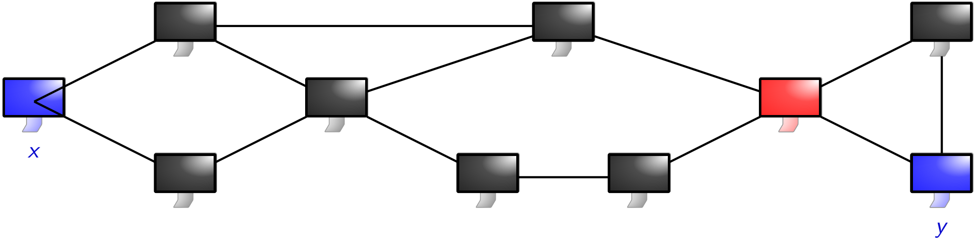
x, y **biconnected** $\iff$ in the same connected component, not separated by another vertex

Biconnectivity



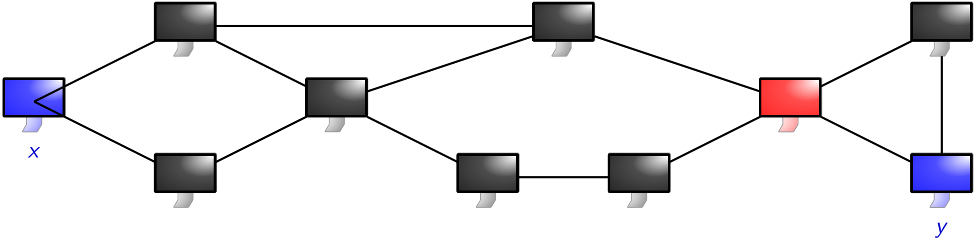
x, y **biconnected** $\iff$ in the same connected component, not separated by another vertex

Biconnectivity



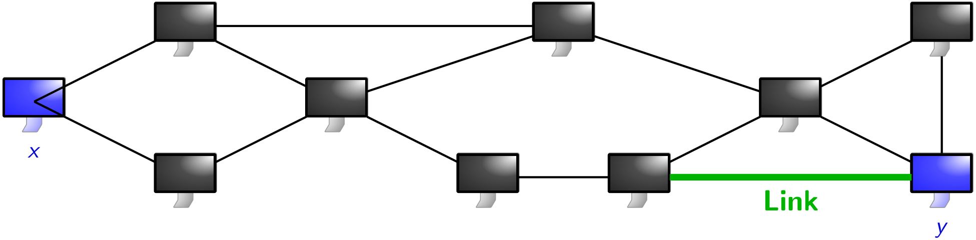
x, y **biconnected** $\iff$ in the same connected component, not separated by **another vertex**

Dynamic Biconnectivity



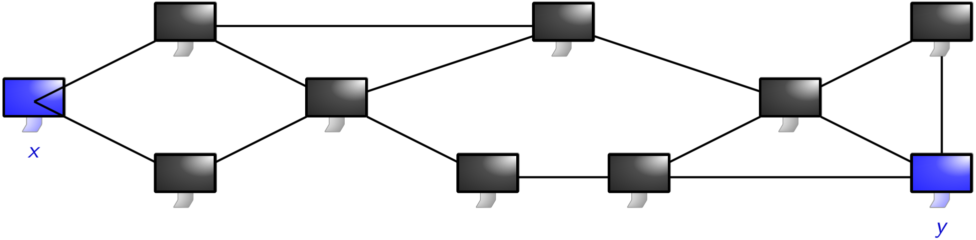
x, y **biconnected** $\iff$ in the same connected component, not separated by **another vertex**

Dynamic Biconnectivity



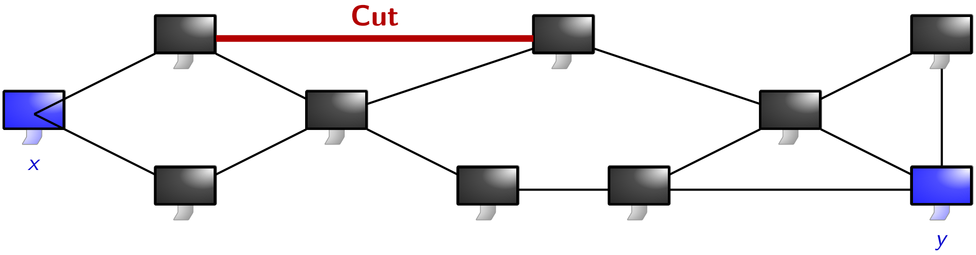
x, y **biconnected** $\iff$ in the same connected component, not separated by another vertex

Dynamic Biconnectivity



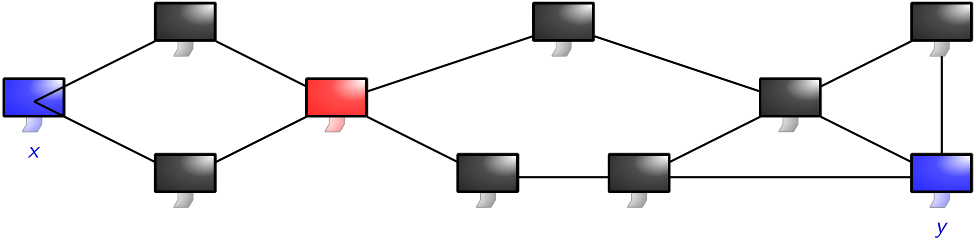
x, y **biconnected** $\iff$ in the same connected component, not separated by another vertex

Dynamic Biconnectivity



x, y **biconnected** $\iff$ in the same connected component, not separated by another vertex

Dynamic Biconnectivity



x, y **biconnected** $\iff$ in the same connected component, not separated by **another vertex**

Dynamic Biconnectivity

Holm, Nadara, Rotenberg, **Sokołowski** [STOC '25]

FULLY DYNAMIC BICONNECTIVITY IN $\widetilde{O}(\log^2 n)$ TIME

Dynamic Biconnectivity

Holm, Nadara, Rotenberg, **Sokołowski** [STOC '25]

FULLY DYNAMIC BICONNECTIVITY IN $\widetilde{O}(\log^2 n)$ TIME

	Update/Query Time	**Deterministic?**
[Henzinger '92]	$\mathcal{O}(m^{2/3})$	yes

Dynamic Biconnectivity

Holm, Nadara, Rotenberg, **Sokołowski** [STOC '25]

Fully Dynamic Biconnectivity in $\widetilde{O}(\log^2 n)$ Time

	Update/Query Time	**Deterministic?**
[Henzinger '92]	$\mathcal{O}(m^{2/3})$	**yes**
[Eppstein et al. '92]	$\mathcal{O}(n^{2/3})$	**yes**

Dynamic Biconnectivity

Holm, Nadara, Rotenberg, **Sokołowski** [STOC '25]

FULLY DYNAMIC BICONNECTIVITY IN $\widetilde{O}(\log^2 n)$ TIME

	Update/Query Time	**Deterministic?**
[Henzinger '92]	$\mathcal{O}(m^{2/3})$	yes
[Eppstein et al. '92]	$\mathcal{O}(n^{2/3})$	yes
[Henzinger '00]	$\mathcal{O}(n^{1/2})$	yes

Dynamic Biconnectivity

Holm, Nadara, Rotenberg, **Sokołowski** [STOC '25]

Fully Dynamic Biconnectivity in $\widetilde{O}(\log^2 n)$ Time

	Update/Query Time	**Deterministic?**
[Henzinger '92]	$\mathcal{O}(m^{2/3})$	**yes**
[Eppstein et al. '92]	$\mathcal{O}(n^{2/3})$	**yes**
[Henzinger '00]	$\mathcal{O}(n^{1/2})$	**yes**
[Henzinger, King '95]	$\mathcal{O}(\log^4 n)$	**no**

Dynamic Biconnectivity

Holm, Nadara, Rotenberg, **Sokołowski** [STOC '25]

FULLY DYNAMIC BICONNECTIVITY IN $\widetilde{\mathcal{O}}(\log^2 n)$ TIME

	Update/Query Time	**Deterministic?**
[Henzinger '92]	$\mathcal{O}(m^{2/3})$	yes
[Eppstein et al. '92]	$\mathcal{O}(n^{2/3})$	yes
[Henzinger '00]	$\mathcal{O}(n^{1/2})$	yes
[Henzinger, King '95]	$\mathcal{O}(\log^4 n)$	no
[Holm, de Lichtenberg, Thorup '98]	$\mathcal{O}(\log^5 n)$	yes

Dynamic Biconnectivity

Holm, Nadara, Rotenberg, **Sokołowski** [STOC '25]

Fully Dynamic Biconnectivity in $\widetilde{\mathcal{O}}(\log^2 n)$ Time

	Update/Query Time	**Deterministic?**
[Henzinger '92]	$\mathcal{O}(m^{2/3})$	yes
[Eppstein et al. '92]	$\mathcal{O}(n^{2/3})$	yes
[Henzinger '00]	$\mathcal{O}(n^{1/2})$	yes
[Henzinger, King '95]	$\mathcal{O}(\log^4 n)$	no
[Holm, de Lichtenberg, Thorup '98]	$\mathcal{O}(\log^5 n)$	yes
[Thorup '00]	$\mathcal{O}(\log^4 n \log\log n)$	yes

Dynamic Biconnectivity

Holm, Nadara, Rotenberg, **Sokołowski** [STOC '25]

Fully Dynamic Biconnectivity in $\widetilde{O}(\log^2 n)$ Time

	Update/Query Time	**Deterministic?**
[Henzinger '92]	$\mathcal{O}(m^{2/3})$	yes
[Eppstein et al. '92]	$\mathcal{O}(n^{2/3})$	yes
[Henzinger '00]	$\mathcal{O}(n^{1/2})$	yes
[Henzinger, King '95]	$\mathcal{O}(\log^4 n)$	no
[Holm, de Lichtenberg, Thorup '98]	$\mathcal{O}(\log^5 n)$	yes
[Thorup '00]	$\mathcal{O}(\log^4 n \log \log n)$	yes
our work	$\mathcal{O}(\log^2 n \log^2 \log n)$	yes

Thank you!

Extra slides: Dynamic Rankwidth

Treewidth, but for denser graphs

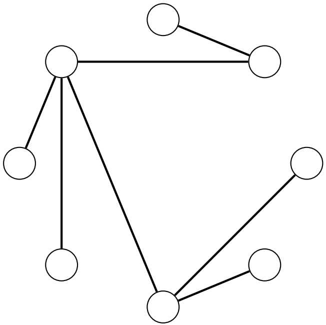
Issue: treewidth applicable only to **sparse** graphs. . .

But there also exist **dense** tree-like graphs!

Treewidth, but for denser graphs

Issue: treewidth applicable only to **sparse** graphs. . .

But there also exist **dense** tree-like graphs!

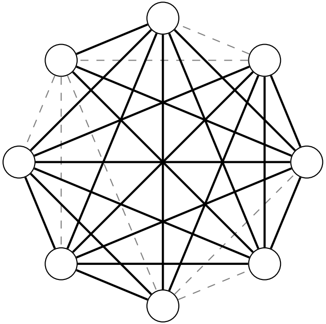


trees

Treewidth, but for denser graphs

Issue: treewidth applicable only to **sparse** graphs. . .

But there also exist **dense** tree-like graphs!

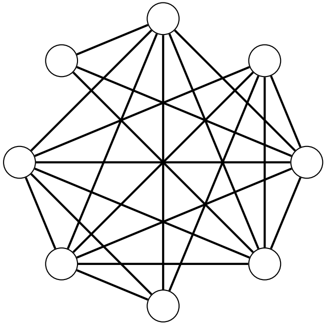


complements of trees

Treewidth, but for denser graphs

Issue: treewidth applicable only to **sparse** graphs. . .

But there also exist **dense** tree-like graphs!

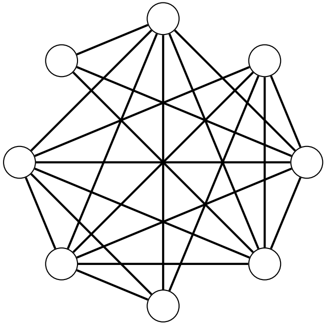


complements of trees

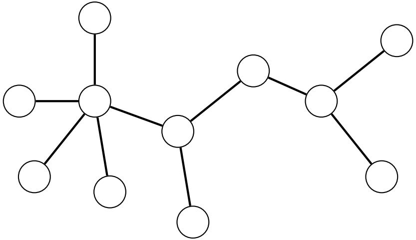
Treewidth, but for denser graphs

Issue: treewidth applicable only to **sparse** graphs. . .

But there also exist **dense** tree-like graphs!



complements of trees

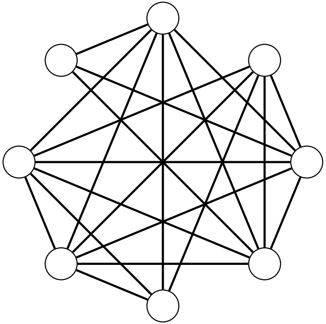


trees

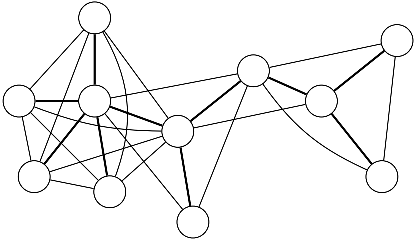
Treewidth, but for denser graphs

Issue: treewidth applicable only to **sparse** graphs. . .

But there also exist **dense** tree-like graphs!



complements of trees



squares of trees

Treewidth, but for denser graphs

Issue: treewidth applicable only to **sparse** graphs. . .

But there also exist **dense** tree-like graphs!

Solution

Equivalent notions of **cliquewidth** [Courcelle et al. '93] and **rankwidth** [Oum, Seymour '06].

Treewidth, but for denser graphs

Issue: treewidth applicable only to **sparse** graphs...

But there also exist **dense** tree-like graphs!

Solution

Equivalent notions of **cliquewidth** [Courcelle et al. '93] and **rankwidth** [Oum, Seymour '06].

Rankwidth is great!

Given: n-vertex graph G and its **rank decomposition** of width w
Then: Maximum Independent Set can be solved in time $2^{f(w)} \cdot n$

Treewidth, but for denser graphs

Issue: treewidth applicable only to **sparse** graphs...

But there also exist **dense** tree-like graphs!

Solution

Equivalent notions of **cliquewidth** [Courcelle et al. '93] and **rankwidth** [Oum, Seymour '06].

Rankwidth is great!

Given: n-vertex graph G and its **rank decomposition** of width w
Then: Maximum Independent Set can be solved in time $2^{f(w)} \cdot n$

Also Max Clique, Min Dominating Set, Longest Induced Path, ...

Rankwidth

Rankwidth is great!

Given: n-vertex graph G and its **rank decomposition** of width w
Then: MAXIMUM INDEPENDENT SET can be solved in time $2^{f(w)} \cdot n$

Same problem: Need to compute a rank decomposition.

Rankwidth

Rankwidth is great!

Given: n-vertex graph G and its **rank decomposition** of width w
Then: MAXIMUM INDEPENDENT SET can be solved in time $2^{f(w)} \cdot n$

Same problem: Need to compute a rank decomposition.

Rank decomposition algorithms

Given an n-vertex graph G of rankwidth w, we can **find** a rank decomposition of G...

	Width guarantee	**Time**
[Oum, Seymour '06]	$3w + 1$	$2^{\mathcal{O}(w)} \cdot n^9$
[Oum '08]	$3w - 1$	$f(w) \cdot n^3$
[Jeong, Kim, Oum '21]	w	$f(w) \cdot n^3$
[Fomin, Korhonen '22]	w	$f(w) \cdot n^2$

Dynamic Rankwidth

Korhonen, **Sokołowski** [STOC '24]

ALMOST-LINEAR TIME PARAMETERIZED ALGORITHM FOR RANKWIDTH VIA DYNAMIC RANKWIDTH

Main result

Dynamic Rankwidth

Korhonen, **Sokołowski** [STOC '24]

ALMOST-LINEAR TIME PARAMETERIZED ALGORITHM FOR RANKWIDTH VIA DYNAMIC RANKWIDTH

Main result

In a **dynamic graph** G with n vertices and m edges of rankwidth w ...

Dynamic Rankwidth

Korhonen, **Sokołowski** [STOC '24]

ALMOST-LINEAR TIME PARAMETERIZED ALGORITHM FOR RANKWIDTH VIA DYNAMIC RANKWIDTH

Main result

In a **dynamic graph** G with n vertices and m edges of rankwidth w ...

We maintain: a rank decomposition of G of width at most $4w$ ...

Dynamic Rankwidth

Korhonen, **Sokołowski** [STOC '24]

ALMOST-LINEAR TIME PARAMETERIZED ALGORITHM FOR RANKWIDTH VIA DYNAMIC RANKWIDTH

Main result

In a **dynamic graph** G with n vertices and m edges of rankwidth $w \ldots$

We maintain: a rank decomposition of G of width at most $4w \ldots$

Initialization time: $2^{f(w)} \cdot n \log^2 n$

Dynamic Rankwidth

Korhonen, **Sokołowski** [STOC '24]

ALMOST-LINEAR TIME PARAMETERIZED ALGORITHM FOR RANKWIDTH VIA DYNAMIC RANKWIDTH

Main result

In a **dynamic graph** G with n vertices and m edges of rankwidth w . . .

We maintain: a rank decomposition of G of width at most $4w$ . . .

Initialization time: $2^{f(w)} \cdot n \log^2 n$

Update time: $2^{f(w) \cdot \sqrt{\log n \log \log n}}$ *(amortized)*

Dynamic Rankwidth

Korhonen, **Sokołowski** [STOC '24]

ALMOST-LINEAR TIME PARAMETERIZED ALGORITHM FOR RANKWIDTH VIA DYNAMIC RANKWIDTH

Main result

In a **dynamic graph** G with n vertices and m edges of rankwidth $w \ldots$

We maintain: a rank decomposition of G of width at most $4w \ldots$

Initialization time: $2^{f(w)} \cdot n \log^2 n$

Update time: $2^{f(w) \cdot \sqrt{\log n \log \log n}}$ *(amortized)*

Extension

We can also dynamically solve any decision/optimization problem expressible in CMSO_1 logic.

Dynamic Rankwidth

Korhonen, **Sokołowski** [STOC '24]

ALMOST-LINEAR TIME PARAMETERIZED ALGORITHM FOR RANKWIDTH VIA DYNAMIC RANKWIDTH

Main result

In a **dynamic graph** G with n vertices and m edges of rankwidth $w \dots$

We maintain: a rank decomposition of G of width at most $4w \dots$

Initialization time: $2^{f(w)} \cdot n \log^2 n$

Update time: $2^{f(w) \cdot \sqrt{\log n \log \log n}}$ *(amortized)*

Extension

We can also dynamically solve any decision/optimization problem expressible in $CMSO_1$ logic.

MAX CLIQUE, MAX INDEPENDENT SET, MIN DOMINATING SET, ~~LONGEST PATH~~...

Dynamic Rankwidth

Rank decomposition algorithms

Given an n-vertex graph G of rankwidth w, we can **find** a rank decomposition of G...

	Width guarantee	**Time**
[Oum, Seymour '06]	$3w + 1$	$2^{\mathcal{O}(w)} \cdot n^9$
[Oum '08]	$3w - 1$	$f(w) \cdot n^3$
[Jeong, Kim, Oum '21]	w	$f(w) \cdot n^3$
[Fomin, Korhonen '22]	w	$f(w) \cdot n^2$

Dynamic Rankwidth

Rank decomposition algorithms

Given an n-vertex graph G of rankwidth w, we can **find** a rank decomposition of G...

	Width guarantee	**Time**
[Oum, Seymour '06]	$3w+1$	$2^{\mathcal{O}(w)} \cdot n^9$
[Oum '08]	$3w-1$	$f(w) \cdot n^3$
[Jeong, Kim, Oum '21]	w	$f(w) \cdot n^3$
[Fomin, Korhonen '22]	w	$f(w) \cdot n^2$
[Korhonen, Sokołowski '24]	w	$f(w) \cdot n^{1+o(1)} + \mathcal{O}(m)$